|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/05 Современные интеллектуальные**

**программно-аппаратные комплексы**

**ОТЧЕТ**

**по лабораторной работе №5**

|  |  |
| --- | --- |
| Группа | ИУ6-23М |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | И.И. Иванов |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Преподаватель |  |  | П.В. Степанов |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

*Москва, 2023 г.*

**Условие**

1. Выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

2. Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

3.9. Входной файл содержит совокупность строк. Строка файла содержит строку квадратной матрицы. Ввести матрицу в двумерный массив (размер матрицы найти). Вывести исходную матрицу и результат ее транспонирования.  
3.10. Входной файл хранит квадратную матрицу по принципу: строка представляет собой число. Определить размерность. Построить 2-мерный массив, содержащий матрицу. Вывести исходную матрицу и результат ее поворота на 90 градусов по часовой стрелке.

4.9. Прочитать строки из файла и поменять местами первое и последнее слова в каждой строке.

4.10. Ввести из текстового файла, связанного с входным потоком, последовательность строк. Выбрать и сохранить m последних слов в каждой из последних n строк

**Програмная реализация**

1.9.

|  |
| --- |
| public class Main {  public static void main(String[] args) {  Furniture furniture = new BookCloset(3);  furniture.display();  ((Closet) furniture).addShelf();  }  } |

1.10.

|  |
| --- |
| import java.io.File;  import java.io.FileNotFoundException;  import java.io.PrintWriter;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  String inputFilePath = "input.txt";  String outputFilePath = "output.txt";  // Если пути к файлам переданы в аргументах командной строки  if (args.length == 2) {  inputFilePath = args[0];  outputFilePath = args[1];  }  File inputFile = new File(inputFilePath);  File outputFile = new File(outputFilePath);  try {  Scanner scanner = new Scanner(inputFile);  int n = 0;  while (scanner.hasNextLine()) {  String line = scanner.nextLine();  String[] elements = line.split("\\s+");  if (elements.length > n) {  n = elements.length;  }  }  double[][] matrix = new double[n][n];  scanner.close();  scanner = new Scanner(inputFile);  int i = 0;  while (scanner.hasNextLine()) {  String line = scanner.nextLine();  String[] elements = line.split("\\s+");  for (int j = 0; j < elements.length; j++) {  matrix[i][j] = Double.parseDouble(elements[j]);  }  i++;  }  scanner.close();  double[][] transposedMatrix = new double[n][n];  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  transposedMatrix[i][j] = matrix[j][i];  }  }  PrintWriter printWriter = new PrintWriter(outputFile);  printWriter.println("Исходная матрица:");  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  printWriter.printf("%.2f ", matrix[i][j]);  }  printWriter.println();  }  printWriter.println();  printWriter.println("Транспонированная матрица:");  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  printWriter.printf("%.2f ", transposedMatrix[i][j]);  }  printWriter.println();  }  printWriter.close();  System.out.println("Результаты работы программы записаны в файл " + outputFilePath);  } catch (FileNotFoundException e) {  System.err.println("Ошибка: файл не найден!");  return;  } catch (NumberFormatException e) {  System.err.println("Ошибка: неверный формат данных в файле!");  return;  }  }  } |

3.9.

|  |
| --- |
| import java.io.File;  import java.io.FileNotFoundException;  import java.io.PrintWriter;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  String inputFilePath = "input.txt";  String outputFilePath = "output.txt";  // Если пути к файлам переданы в аргументах командной строки  if (args.length == 2) {  inputFilePath = args[0];  outputFilePath = args[1];  }  File inputFile = new File(inputFilePath);  File outputFile = new File(outputFilePath);  try {  Scanner scanner = new Scanner(inputFile);  int n = 0;  while (scanner.hasNextLine()) {  String line = scanner.nextLine();  String[] elements = line.split("\\s+");  if (elements.length > n) {  n = elements.length;  }  }  double[][] matrix = new double[n][n];  scanner.close();  scanner = new Scanner(inputFile);  int i = 0;  while (scanner.hasNextLine()) {  String line = scanner.nextLine();  String[] elements = line.split("\\s+");  for (int j = 0; j < elements.length; j++) {  matrix[i][j] = Double.parseDouble(elements[j]);  }  i++;  }  scanner.close();  double[][] transposedMatrix = new double[n][n];  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  transposedMatrix[i][j] = matrix[j][i];  }  }  PrintWriter printWriter = new PrintWriter(outputFile);  printWriter.println("Исходная матрица:");  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  printWriter.printf("%.2f ", matrix[i][j]);  }  printWriter.println();  }  printWriter.println();  printWriter.println("Транспонированная матрица:");  for (i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  printWriter.printf("%.2f ", transposedMatrix[i][j]);  }  printWriter.println();  }  printWriter.close();  System.out.println("Результаты работы программы записаны в файл " + outputFilePath);  } catch (FileNotFoundException e) {  System.err.println("Ошибка: файл не найден!");  return;  } catch (NumberFormatException e) {  System.err.println("Ошибка: неверный формат данных в файле!");  return;  }  }  } |

3.10.

|  |
| --- |
| import java.io.File;  import java.io.IOException;  import java.util.Scanner;  import java.io.PrintWriter;  public class Main {  public static void main(String[] args) throws IOException {  // Создаем объект File для файла, который хотим прочитать  File inputFile = new File("input.txt");  // Создаем объект Scanner для чтения данных из файла  Scanner scanner = new Scanner(inputFile);  // Создаем объект File для создания нового файла  File outputFile = new File("out/output.txt");  // Создаем директорию out, если ее нет  File outDir = new File("out");  outDir.mkdir();  // Создаем объект PrintWriter для записи данных в новый файл  PrintWriter writer = new PrintWriter(outputFile);  // Читаем строки из файла и меняем местами первое и последнее слова  while (scanner.hasNextLine()) {  String line = scanner.nextLine();  String[] words = line.split("\\s+");  if (words.length > 1) {  String firstWord = words[0];  String lastWord = words[words.length - 1];  words[0] = lastWord;  words[words.length - 1] = firstWord;  }  String newLine = String.join(" ", words);  writer.println(newLine);  }  // Закрываем объекты Scanner и PrintWriter  scanner.close();  writer.close();  }  } |